DTU Risø DTU
National Laboratory for Sustainable Energy

McXtrace

McXtrace - An X-ray Monte Carlo Ray-tracing software package

Erik Knudsen[1], Peter Willendrup[1], Søren Schmidt[1], Kim Lefmann[2], Jana Baltser[2], Andrea Prodi[2], Søren Kynde[3], Kell Mortensen[3]

[1]RISØ DTU, Frederiksborgvej 399, DK-4000 Roskilde
[2]Niels Bohr Institute (NBI), Universitetsparken 5, DK-2100 Copenhagen, Denmark
[3]Faculty of Life Sciences, Copenhagen University (KU-LIFE), Bülowsvej 17, DK-1870 Frederiksberg C, Denmark

Abstract

We present the prototype release of the McXtrace Monte Carlo X-ray tracing software package. The package has been derived from the successful and proven McStas package for neutron ray tracing. The software package, realised as an open-source project, is aimed at simulating X-ray beam lines and instrumentation, in short to answer questions such as "Should we buy device X or Y to improve our data quality?" or "What are the optimal parameter settings of device Z?"

For starters, we discuss the differences between neutron and X-ray transport, how the two concepts are interlinked and how codes describing either may benefit from each other.

We show early results from a couple of simpler instrumental setups — examples include a model of an laboratory SAXS-setup and a low-budget monochromatic beam line, and a discussion about the efficiency of the code. Furthermore, the open structure of the code is presented and how users may contribute, extend and benefit from it, for instance by interfacing to already existing, proven tools.

Conversion from McStas

The code structure of McXtrace, inherited directly from McStas, is basically divided into 3 parts:

Instrument Consists of Components. May simply be a list of components in the setup with their coordinates in the McStas-meta language, but may also include complex programming. Users are required to write such a file - assisted by GUI/helper applications.

Component Where "physics" happen. Inside components the interactions between neutrons/X-rays and matter are described. Some users who wish to contribute and/or develop new features (not yet in the library components) write these. Often components are initiated by users and completed and maintained by the McStas/McXtrace team. As components are usually no more than ≈ 300 lines of code, this is usually a manageable task, even for non-specialized programmers[a].

Kernel The Kernel takes care of positioning the bits and pieces of an neutron/X-ray experimental setup and handles beam transport. No users need to bother about this.

[a]such as physicists

In the same manner, the process of spinning off McXtrace from McStas occurs on the same three levels. On the Kernel level the transport mechanisms are edited to reflect the properties of an X-ray photon vs. a neutron.

Neutron	X-ray photon
$\mathrm{n} = (\mathbf{r}, \mathbf{v}, p, t, \mathbf{s})$	$\tilde{\mathrm{p}} = (\mathbf{r}, \mathbf{k}, p, t, \phi, \mathbf{E})$
→ time domain propagation	→ freq. domain propagation
• magnetic field/spin polarization	• optical polarization
• gravitational field	• phase propagation/wave optics

Ray-tracing vs. wave optics

Seen from the perspective of a single neutron/x-ray photon, the particle travels from device to device, or in the language of McStas/McXtrace *from component to component*, where the kernel takes care of transport between components. The Kernel itself sees components as "black boxes". Within a component, code (generally written in c) may do anything to the particle. One upshot of this is that inside a component, indeed a different mode of transport may be employed. For instance, inside the single crystal component the ray is considered to be a wave which interacts with the crystal and exits the component as a (possibly) modified ray.

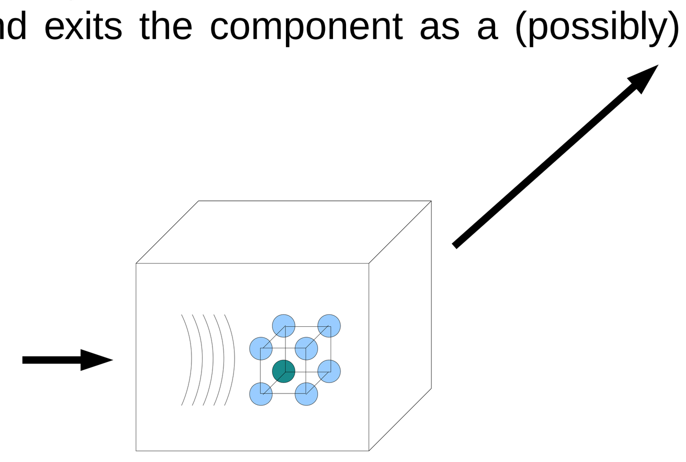
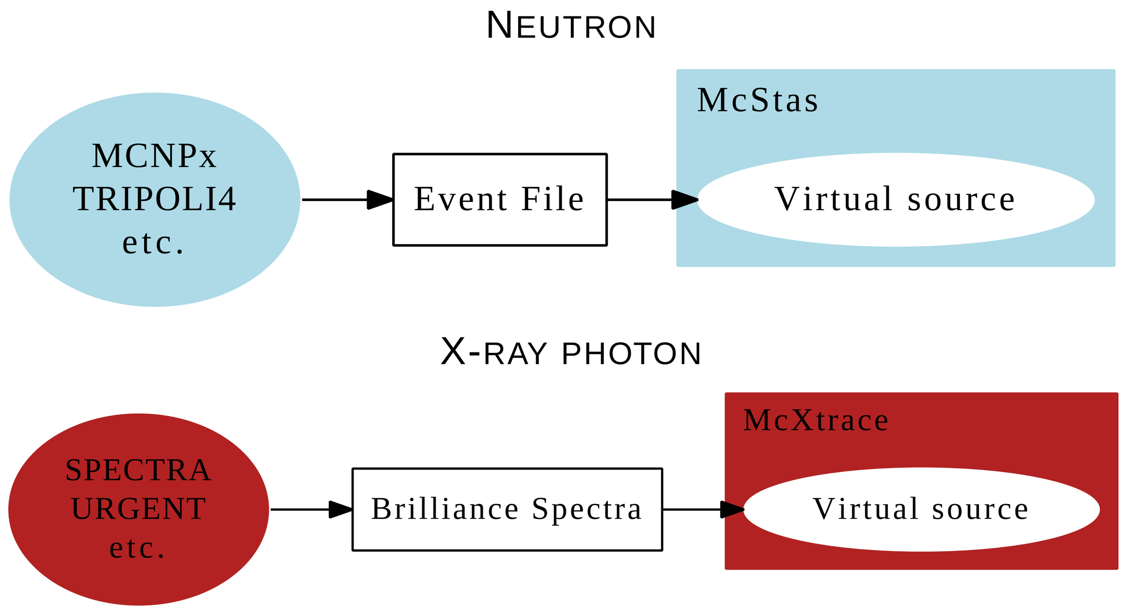


Figure 1: Schematic of the Single crystal component in McStas/McXtrace. A ray enters the component, inside the component it is treated as a wave and, on exit, is converted back to a ray again

Tie-ins with established software packages

General philosophy: If someone does it well already - interface, don't reinvent *but* have very simple models in-house. In terms of sources for instance the structure is:



Another example of this kind of interfacing is the single crystal sample component where input generally comes from a crystallographic program such as "Crystallographica"[3].

1st batch of Target instruments

A set of three target instruments have been set for the first phase of the project with increased complexity as we go along.

Low Budget Monochromatic Beamline

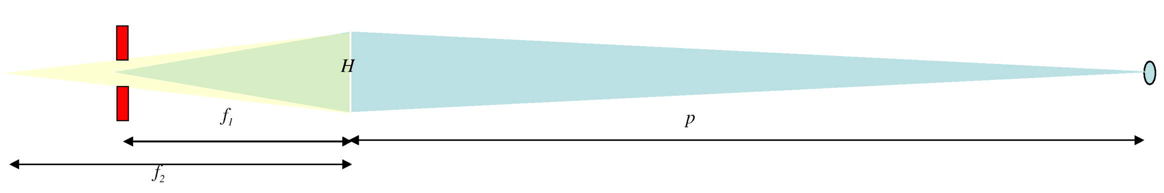
The 1st target instrument is a proposed Low-cost monchromatic beamline, where monochromaticity is provided by a simple slit and a Be-lens. The slit is set in the focal point of the lens for the targeted wavelength, λ_0. In principle, wavelengths $\neq \lambda_0$ will be absorbed by the slit. Due to its simplicity this kind of arrangement would be very cheap compared to other, more sofisticated solutions.

If we denote the lens aperture H, the source size σ, the image size in the focal point becomes: $h = (f_1/p)\sigma$. Further, for the lens:

$$\lambda_1 f_1 = \lambda_2 f_2 = c^2$$

Thus, we may write the energy resolution as:

$$\frac{h}{H} = \frac{f_2 - f_1}{f_2} = \frac{\frac{c^2}{\lambda_2} - \frac{c^2}{\lambda_1}}{\frac{c^2}{\lambda_2}} = \frac{E_2 - E_1}{E_2} = \frac{\Delta E}{E} \Rightarrow \frac{\Delta E}{E} = \frac{f_1}{p}\frac{\sigma}{H}$$



As example: given

$$\frac{f_1}{p} = \frac{1}{10};\ H = 0.8mm;\ \sigma = 0.08mm;$$

we get

$$\frac{\Delta E}{E} = 1\%$$

This very simple setup may be readily modelled using McXtrace. The Lens model is based on [4] and a Gaussian cross-section approximation is applied to the source.

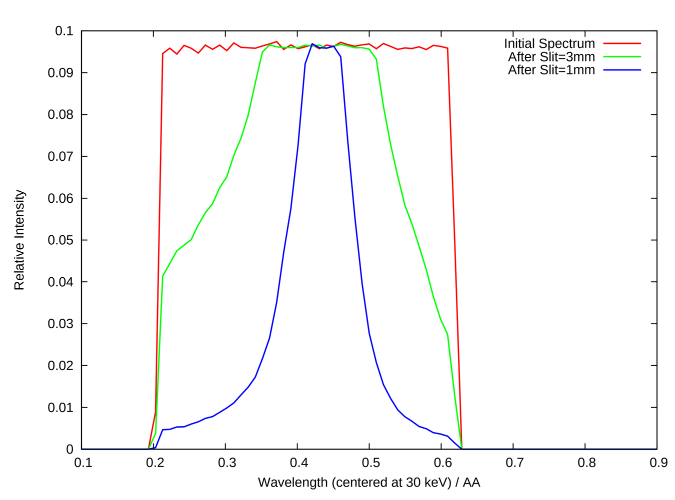


Figure 2: Spectra after "slit"-monochromator

SAXS

Laboratory equipment may also be simulated using this approach. A collaboration agreement is setup with an external company to model their Small Angle Xray Scattering product.[a]

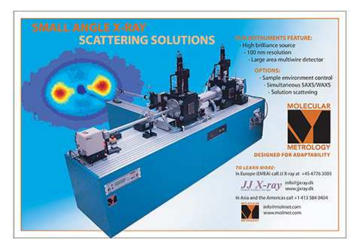


Figure 3: Picture of a SAXSLAB machine.

A very crude model has been produced to model this setup using a flat surface source model, a cylindrical bent mirror, and a set of pinholes, as a proof of concept model.

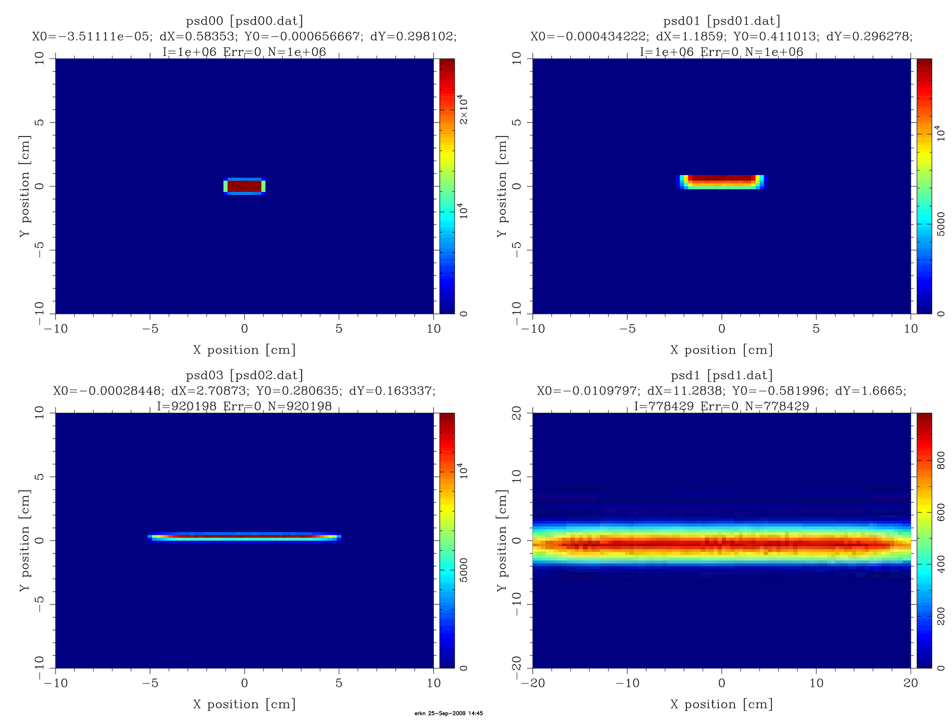


Figure 4: Evolution of beam in a SAXS-type instrument simulation. Beam intensity cross-sections at: top left) just before optic, top right) after 1st pinhole, bottom left) after 3nd pinhole, bottom right) at end detector

[a]The 1st fundamental milestone of the McXtrace project entails a detailed simulation of this setup

ESRF, ID11

The insertion device Beamline ID11 at the European Synchrotron Radiation Facility[5], hosts the 3DXRD-microscope[6].[a] It is a well chosen target instrument since it consists of a fairly complex setup with many different optical options, yet one where the different objects may easily be thought of as discrete entities or components.

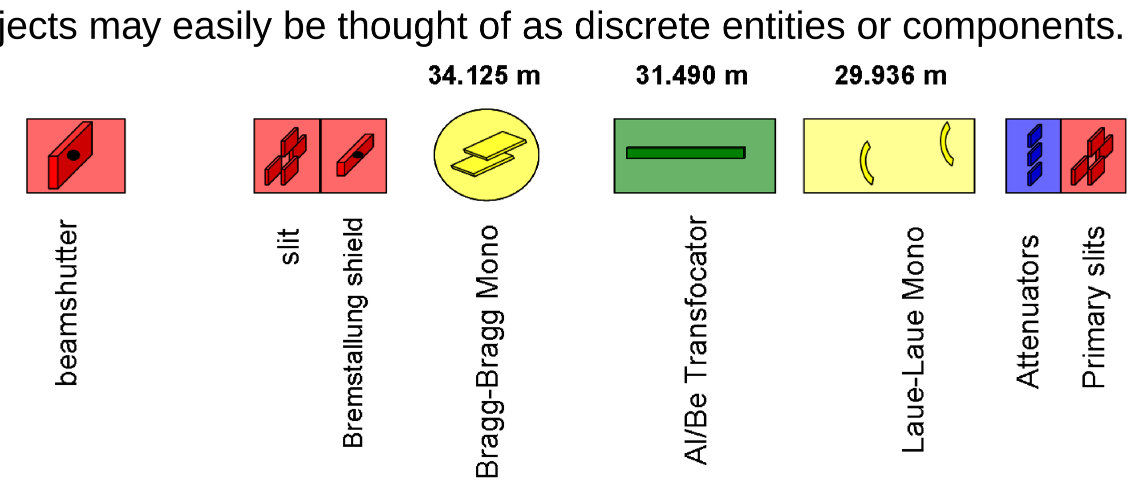


Figure 5: Schematic drawing of the primary optics elements of beamline ID11 (ESRF).

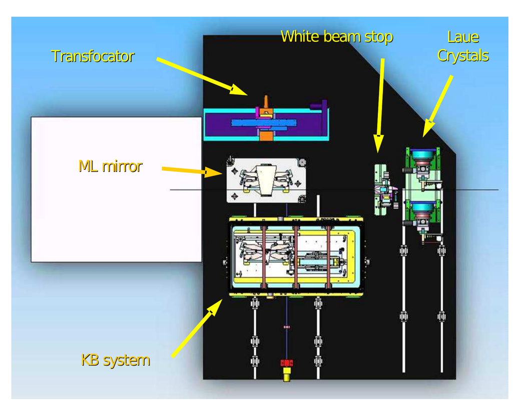


Figure 6: Schematic drawing of the secondary optics elements of beamline ID11 (ESRF).

In addition to this we need several different types of area detector components, as well as various types of sample components, including polycrystals.

[a]The 2nd fundamental milestone of the McXtrace project entails a detailed simulation of this setup

References

[1] *McStas* project website at http://www.mcstas.org

[2] *McXtrace* project website at http://www.mcxtrace.org

[3] *Crystallographica* website at http://www.crystallographica.com

[4] Snigirev, A. et.al. , *Nature*, vol. 384, pp. 49, 1996

[5] *ESRF* website http://www.esrf.eu

[6] H.F. Poulsen, *Three-Dimensional X-Ray Diffraction Microscopy*, Springer Verlag, 2004.

Keywords: McXtrace, McStas, Monte Carlo sim., Software, X-ray, Neutrons, Instruments *PACS:* 61.12-q, 05.10.Ln